Informed Search Strategies

Chapter 3

NORTH-WEST UNIVERSITY
YUNIBESITI YA BOKONE-BOPHIRIMA
NOORDWES-UNIVERSITEIT
INSTITUTIONAL OFFICE

Overview of lecture

- Uninformed search strategies very ineffective
- Consider informed search strategies that find solutions more effective
- Use problem specific knowledge

Informed search strategies

- Informed search strategy — that uses domain-specific hints about the location of goals — can find solutions more efficiently than an uninformed strategy
- The hints come in the form of a heuristic function, denoted by *h(n)*
- *h(n)* = estimated cost of the cheapest path from the state at node *n* to a goal state
- In route-finding problems, we use straight-line distance on the map between the two points

Greedy best-first search

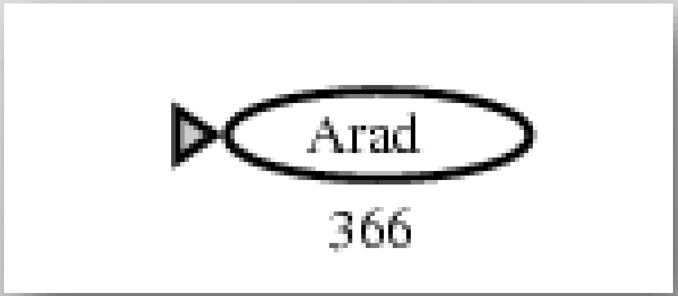
- Greedy best-first search is a form of best-first search that expands the node with the lowest *h(n)* value first
- This node is likely to lead to a solution quickly
- The evaluation function $f(n) = h(n)$
- For the route-finding problems in Romania; we use the straight-line distance heuristic, which we will call h_{SLD}

Greedy best-first search

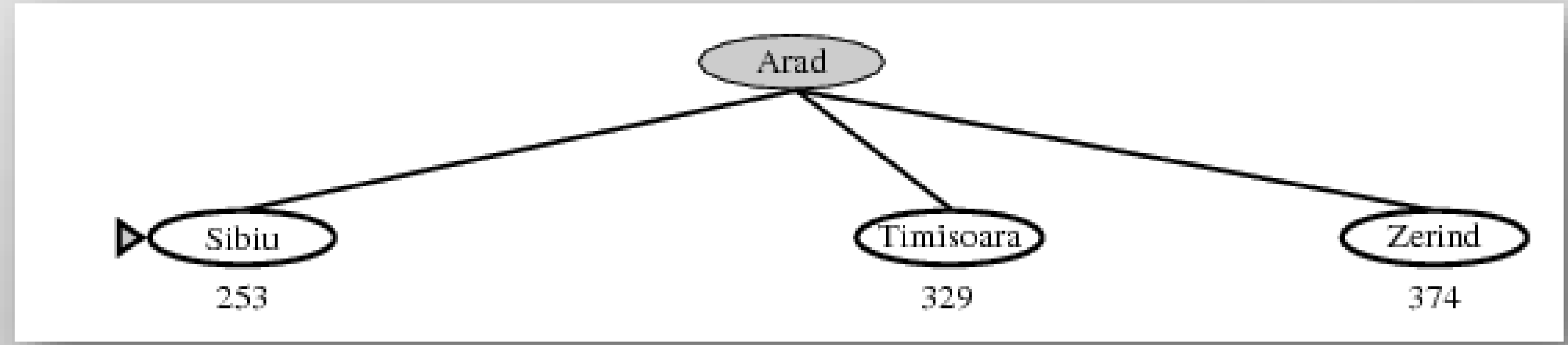
Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244

Mehadia	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

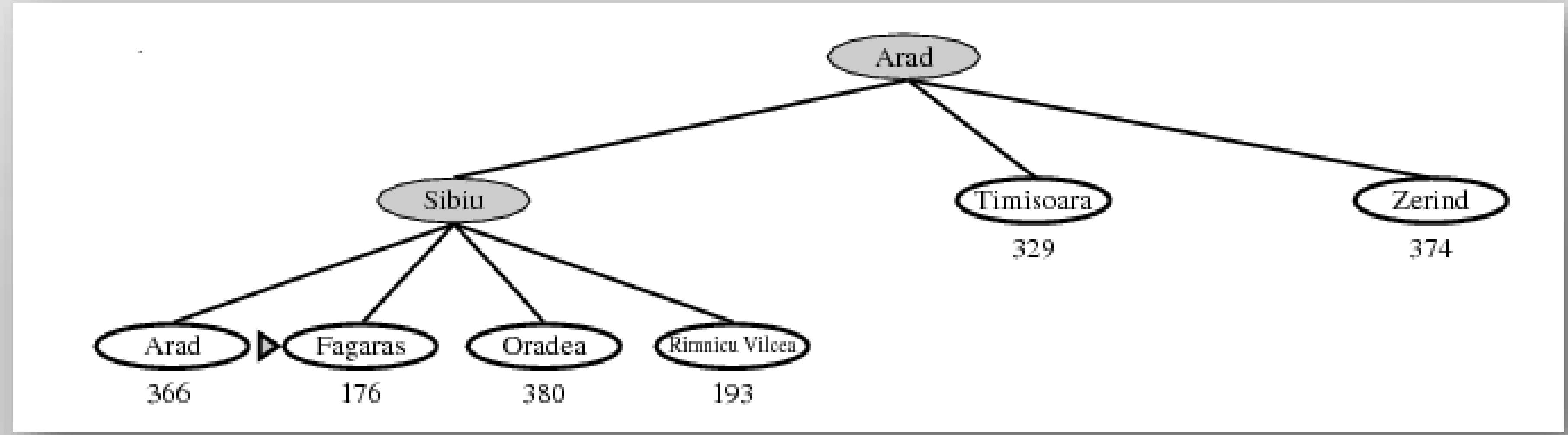
Greedy best-first search



Greedy best-first search

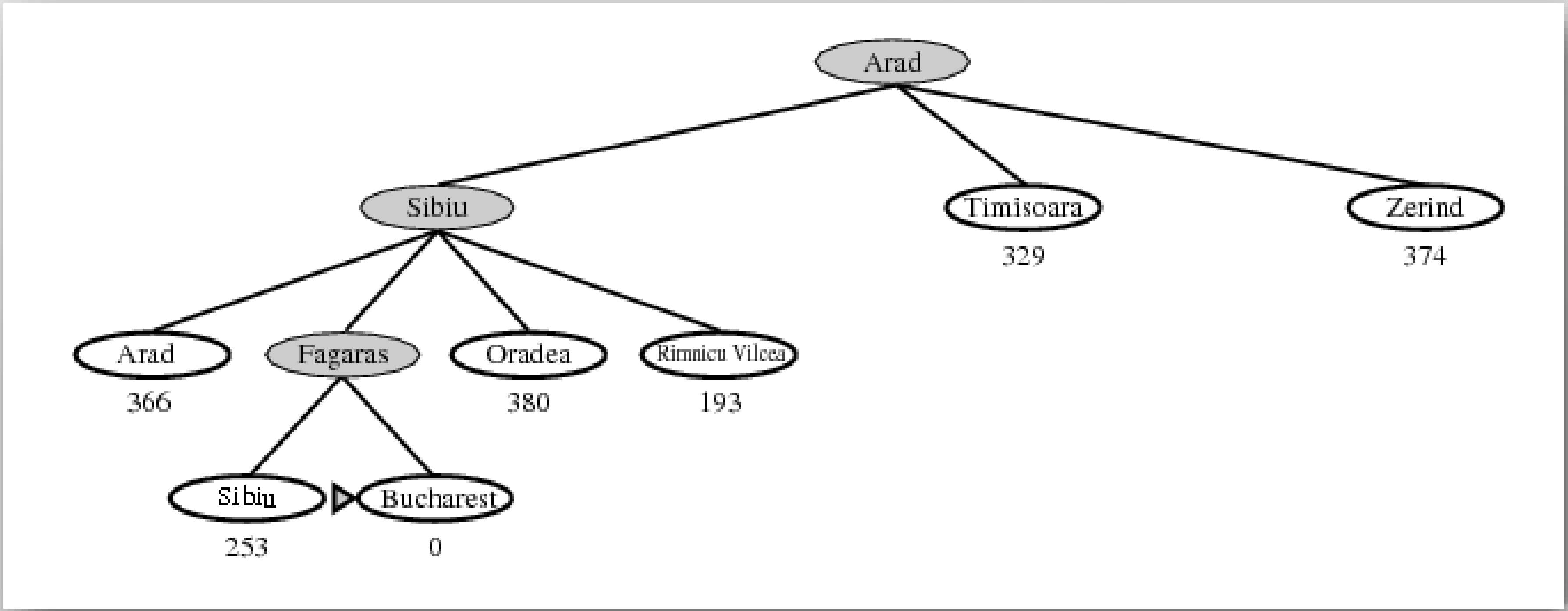


Greedy best-first search

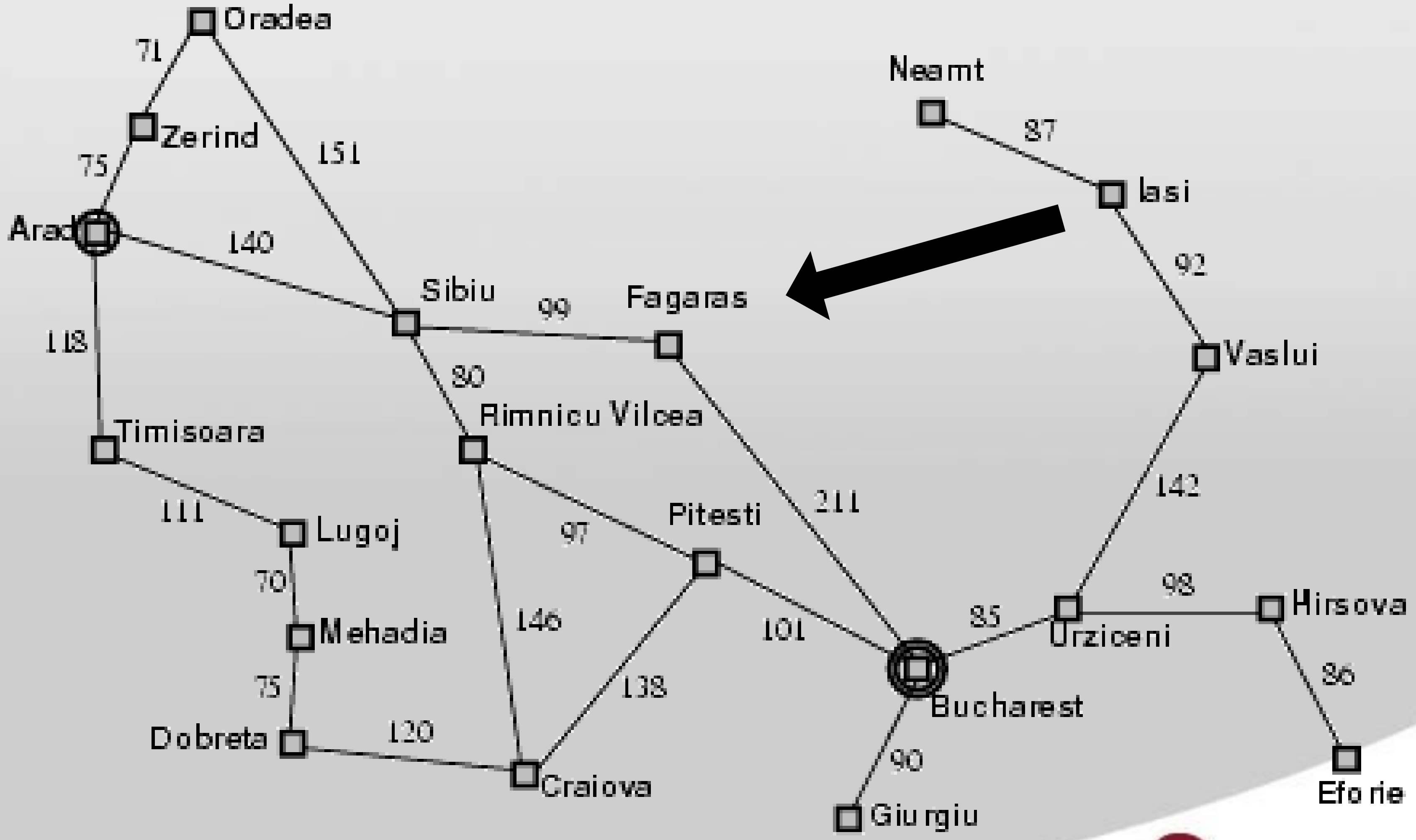


NORTH-WEST UNIVERSITY
YUNIBESITI YA BOKONE-BOPHIRIMA
NOORDWES-UNIVERSITEIT
INSTITUTIONAL OFFICE

Greedy best-first search



Greedy best-first search



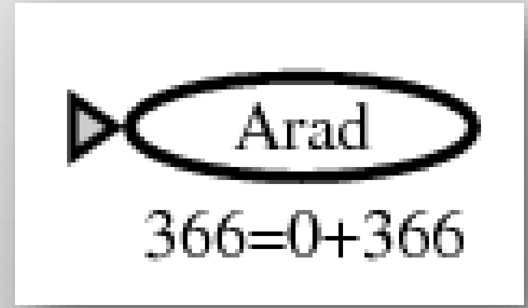
Greedy best-first search

- Greedy best-first graph search is complete in finite state spaces, but not in infinite ones
- The worst-case time and space complexity is $O(|V|)$
- With a good heuristic function, however, the complexity can be reduced substantially, on certain problems reaching $O(bm)$

A* search

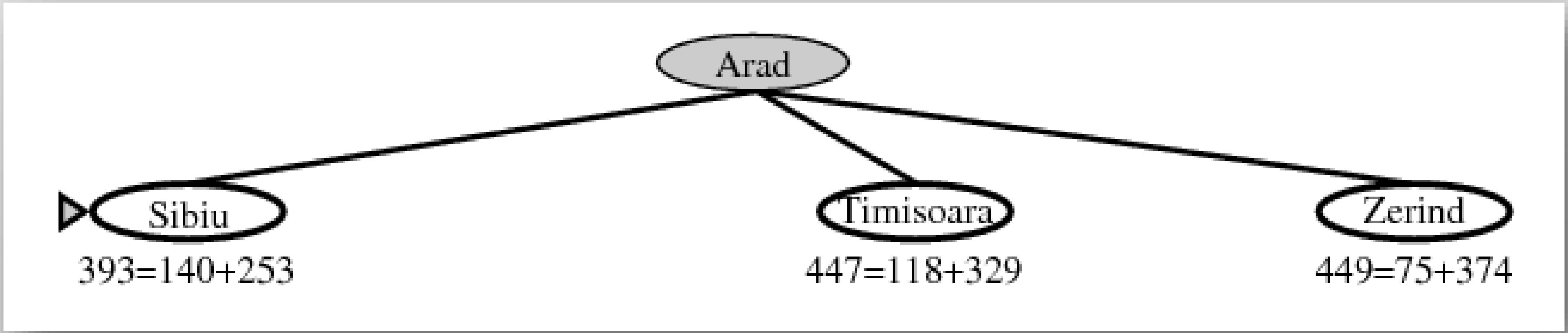
- Most common informed search algorithm is called A* search
- It is a best-first search that uses the evaluation function *f(n)* = *g(n)* + *h(n)*
- where *g(n)* is the path cost from the initial state to node *n* and *h(n)* is the estimated cost of the shortest path from *n* to a goal state, and *f(n)* = estimated cost of the best path that continues from *n* to a goal

A* search



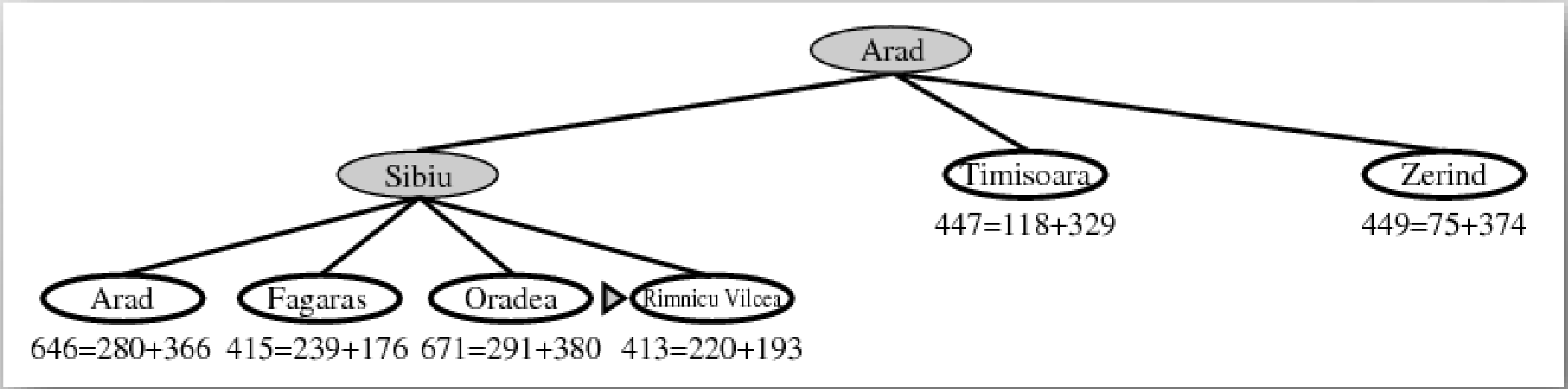
$$f(n) = g(n) + h(n)$$

A* search



$$f(n) = g(n) + h(n)$$

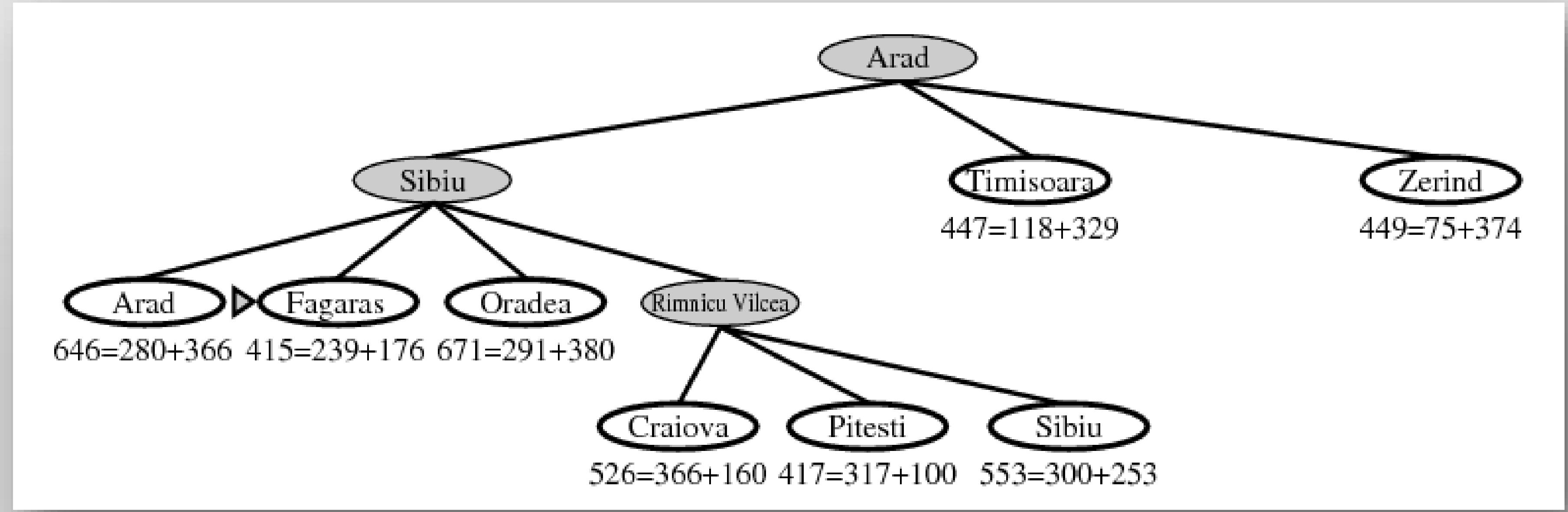
A* search



$$f(n) = g(n) + h(n)$$

NORTH-WEST UNIVERSITY
YUNIBESITI YA BOKONE-BOPHIRIMA
NOORDWES-UNIVERSITEIT
INSTITUTIONAL OFFICE

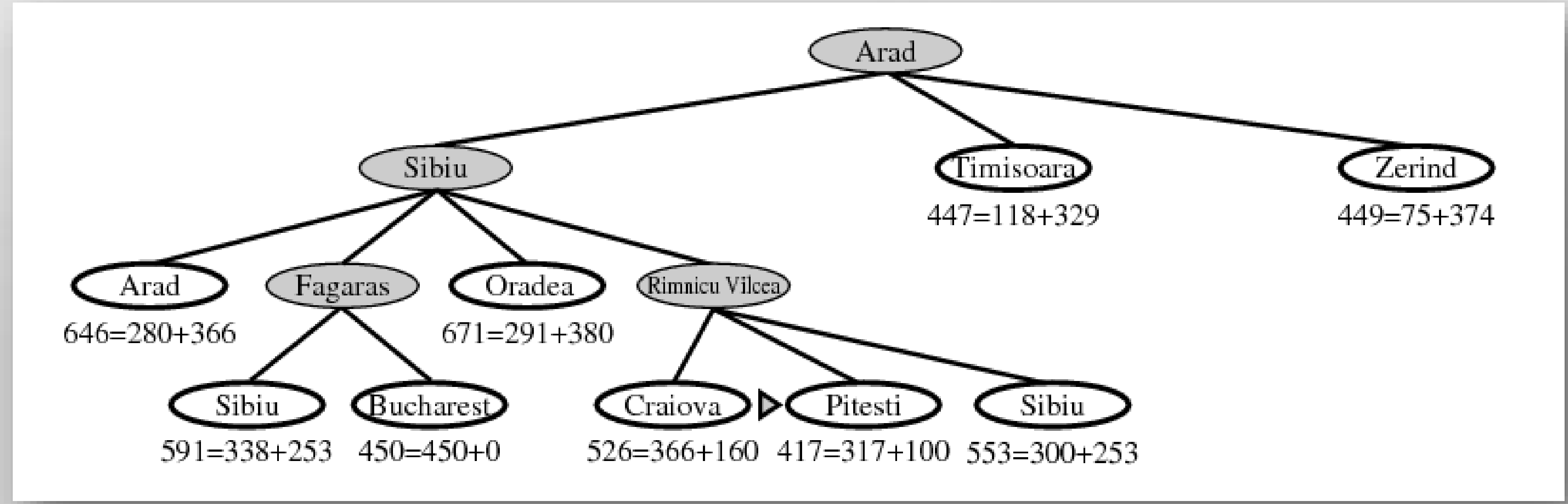
A* search



$$f(n) = g(n) + h(n)$$

NORTH-WEST UNIVERSITY
YUNIBESITI YA BOKONE-BOPHIRIMA
NOORDWES-UNIVERSITEIT
INSTITUTIONAL OFFICE

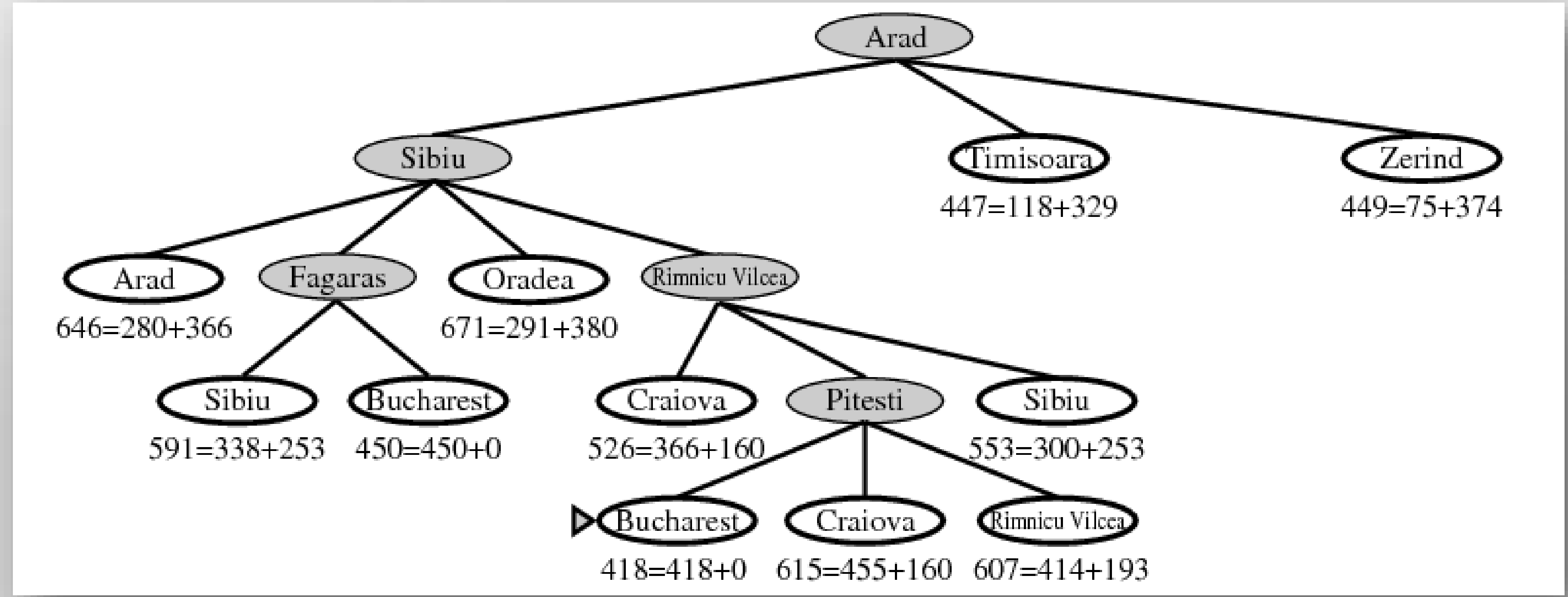
A* search



$$f(n) = g(n) + h(n)$$

NORTH-WEST UNIVERSITY
YUNIBESITI YA BOKONE-BOPHIRIMA
NOORDWES-UNIVERSITEIT
INSTITUTIONAL OFFICE

A* search



$$f(n) = g(n) + h(n)$$

A* search

- A* search is complete
- Whether A* is cost-optimal depends on certain properties of the heuristic
- A key property is admissibility: an admissible heuristic is one that never overestimates the cost to reach a goal
- An admissible heuristic is therefore optimistic
- With an admissible heuristic, A* is cost-optimal

NORTH-WEST UNIVERSITY
YUNIBESITI YA BOKONE-BOPHIRIMA
NOORDWES-UNIVERSITEIT
INSTITUTIONAL OFFICE

Heuristic characteristics

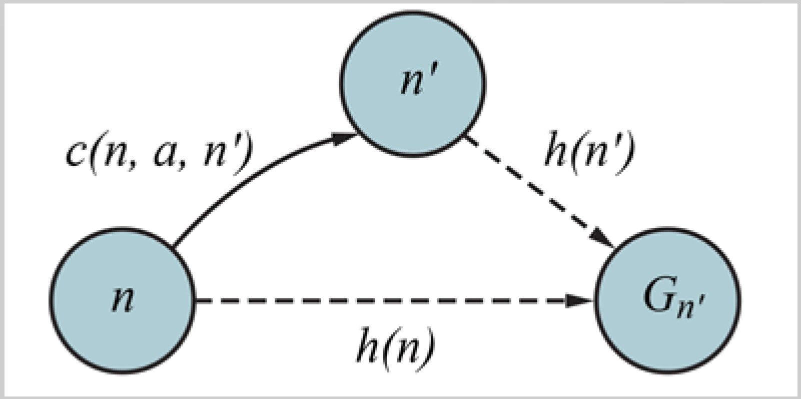
- With an admissible heuristic, A* is cost-optimal, but a stronger indicator of optimality is whether the heuristic is consistent.
- For consistency, the heuristic should always follow this rule:

$$h(n) \leq c(n, a, n') + h(n')$$

Consistency

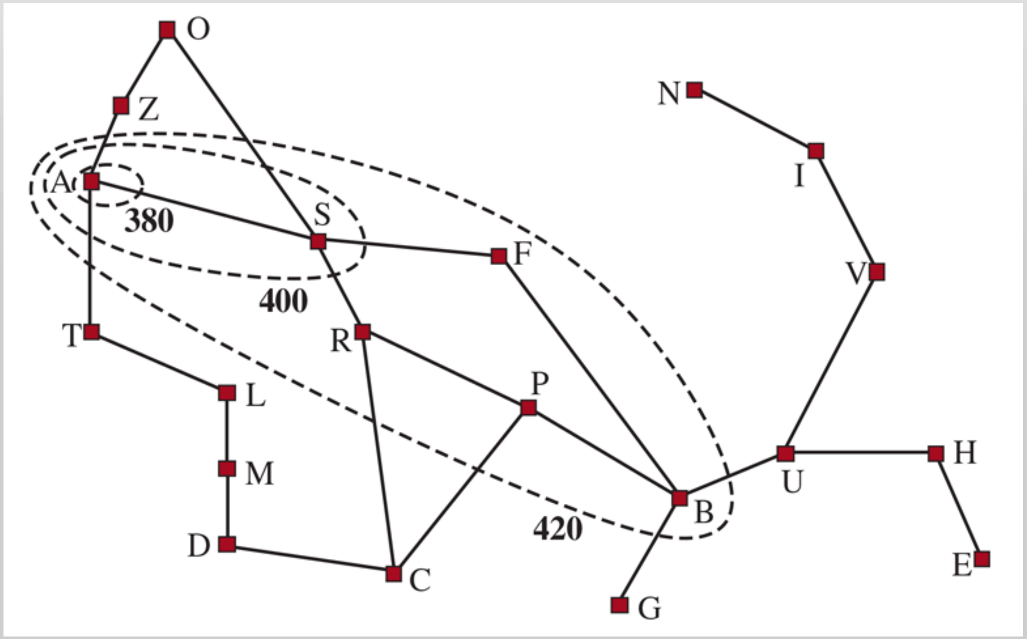
$$h(n) \leq c(n, a, n') + h(n')$$

- This is a type of triangle inequality which states that two sides of a triangle can never be shorter than the other side



Search Contours

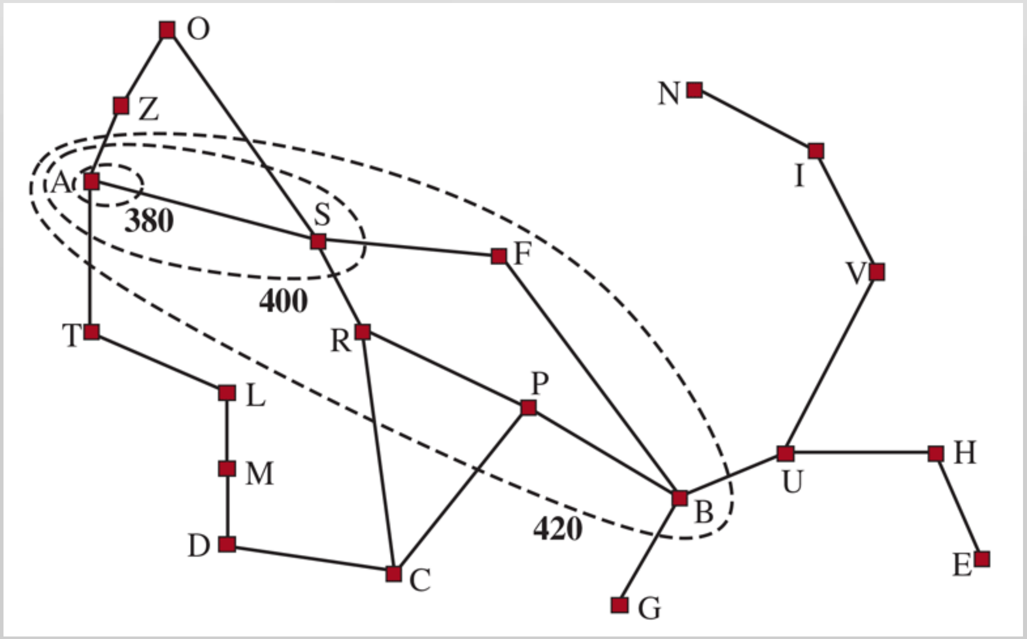
- A way to visualize a search is by viewing it in terms of contours that expand out from start towards the goal



NORTH-WEST UNIVERSITY
YUNIBESITI YA BOKONE-BOPHIRIMA
NOORDWES-UNIVERSITEIT
INSTITUTIONAL OFFICE

Search Contours

- For each of the nodes inside of the contour $f(n) = g(n) + h(n) \leq contour$



NORTH-WEST UNIVERSITY
YUNIBESITI YA BOKONE-BOPHIRIMA
NOORDWES-UNIVERSITEIT
INSTITUTIONAL OFFICE

Search Contours

- Uniform cost search will also generate contours, but in that case the contours will only be $f(n) = g(n)$, creating concentric circles.
- If the action costs are always positive, then we say that g is monotonic – the path grows longer as you add more actions
- In A* search this is more complex though as it is not clear that $f = h+g$ will increase monotonically…

NORTH-WEST UNIVERSITY
YUNIBESITI YA BOKONE-BOPHIRIMA
NOORDWES-UNIVERSITEIT
INSTITUTIONAL OFFICE

Search Contours

- If you extend a path from n to n’ the cost goes from

$$f(n) = g(n) + h(n)$$

to

$$f(n') = g(n) + c(n, a, n') + h(n')$$

- Eliminating g(n), we see that the cost will only increase monotonically if:

$$h(n) \leq c(n, a, n') + h(n')$$

- In other words, only if the heuristic is consistent!

NORTH-WEST UNIVERSITY
YUNIBESITI YA BOKONE-BOPHIRIMA
NOORDWES-UNIVERSITEIT
INSTITUTIONAL OFFICE

A* and Search Contours

- A* expands all nodes that can be reached from the initial state with $f(n) < C*$ these are called surely expanded nodes
- A* might expand some nodes on the goal contour where $f(n) = C*$
- A* expands no nodes where $f(n) > C*$
- A* with a consistent heuristic is optimally efficient

NORTH-WEST UNIVERSITY
YUNIBESITI YA BOKONE-BOPHIRIMA
NOORDWES-UNIVERSITEIT
INSTITUTIONAL OFFICE

Satisficing search

- A* has good qualities but still expands a lot of nodes.
- We can optimise it more if we are willing to accept sub-optimal solutions, also called satisficing solutions.
- If we use inadmissible heuristics we can use a heuristic that is more accurate (not as optimistic) which leads to less nodes being expanded

Satisficing search

- A* has good qualities but still expands a lot of nodes.
- We can optimise it more if we are willing to accept sub-optimal solutions, also called satisficing solutions.
- If we use inadmissible heuristics we can use a heuristic that is more accurate (not as optimistic) which leads to less nodes being expanded

Satisficing search

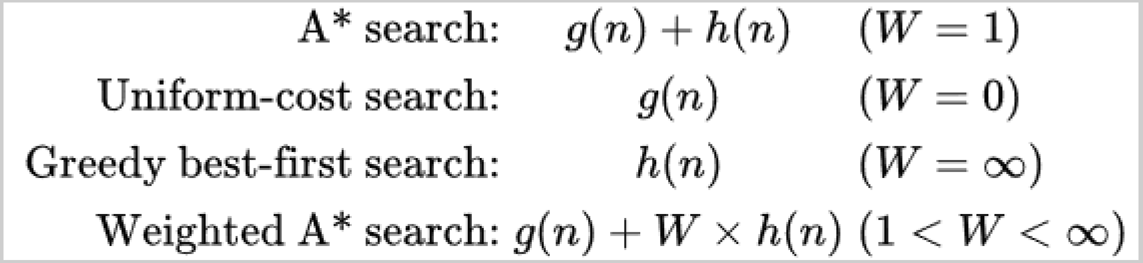
- For example, road engineers know the concept of a detour index, which is a multiplier applied to the straight-line distance to account for the typical curvature of roads. A detour index of 1.3 means that if two cities are 10 miles apart in straight-line distance, a good estimate of the best path between them is 13 miles. For most localities, the detour index ranges between 1.2 and 1.6.

Satisficing search

- If we apply such a heuristic to search we are performing a weighted A* search.
- The evaluation function for Weighted A* is therefor:

$$f(n) = g(n) + W \times h(n)$$

- Where W is the weighting value with $W > 1$



A* search:	$g(n) + h(n)$	$(W = 1)$
Uniform-cost search:	$g(n)$	$(W = 0)$
Greedy best-first search:	$h(n)$	$(W = \infty)$
Weighted A* search:	$g(n) + W \times h(n)$	$(1 < W < \infty)$

NORTH-WEST UNIVERSITY
YUNIBESITI YA BOKONE-BOPHIRIMA
NOORDWES-UNIVERSITEIT
INSTITUTIONAL OFFICE

Suboptimal Search

- Bounded suboptimal search looks for a solution that is within a constant factor of W of the optimal cost. Weighted A* provides this guarantee.
- In Bounded-cost search the solution should be less than some constant C.

Suboptimal Search

- In unbounded-cost search we accept solutions of any cost as long as we can find them quickly.
- An example of an unbounded-cost search algorithm is speedy search, which is a version of greedy best-first search that uses as a heuristic the estimated number of actions required to reach a goal, regardless of the cost of those actions

NORTH-WEST UNIVERSITY
YUNIBESITI YA BOKONE-BOPHIRIMA
NOORDWES-UNIVERSITEIT
INSTITUTIONAL OFFICE

Memory Bounded Search

- The main issue with A* is its use of memory.
- Search algorithms that try to overcome this are for example beam search that limits the size of the frontier nodes to those with the best f-scores.
- Iterative Deepening A* (IDA*) successively increases the f-cost to create successively deeper contours.

Memory Bounded Search

- Simplified Memory-Bounded A* (SMA*) expands the best leaves in the frontier until the memory allocated to it is full, and then drops the worst leaf nodes from memory as it progresses.
- This has the limitation that if the goal node path length is larger than memory then it cannot reach the goal.
- SMA* is complete if depth d of the shallowest goal node is less than the memory limit.

Bidirectional Heuristic Search

- With unidirectional search A* with an admissible (or consistent) heuristic gives us an optimal path.
- This however doesn’t work for bidirectional search as you need nodes from both frontiers to find each other in the middle.

(Study the math and example of bidirectional heuristic search in your textbook)

Announcements

- Study: Chapter 3.5 (Informed (Heuristic) Search Strategies) of the AIMA e-book
- Self-study: Chapter 6 (Building your first deep neural network) of the Grokking Deep Learning e-book, pages 99 – 115

Announcements

- Test 2: Chapter 3.2 (Example Problems) to Chapter 3.5 (Informed (Heuristic) Search Strategies) of the AIMA e-book
  - Thursday, 3 June 2021
- Theory Quiz 7: Chapter 3.5 (Informed (Heuristic) Search Strategies) of the AIMA e-book
  - Thursday, 10 June 2021